System and process for enhancing method calls of special purpose object-oriented programming languages to have security attributes for access control

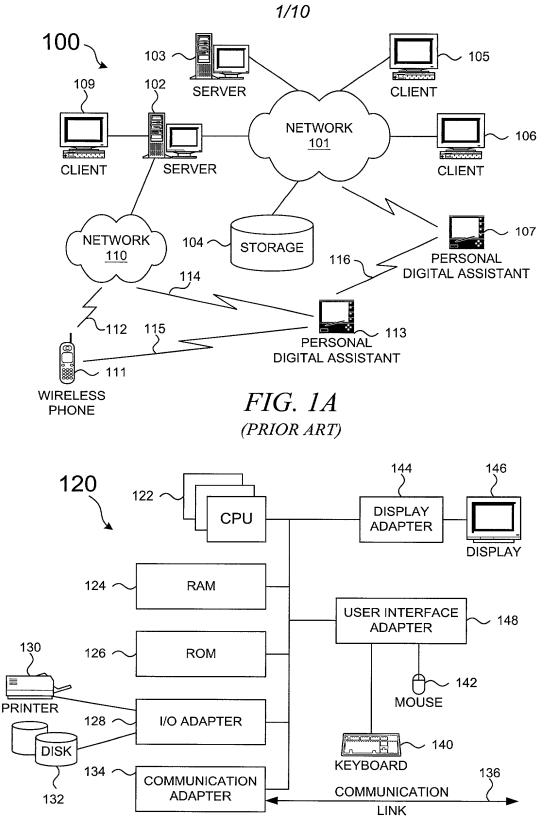


FIG. 1B (PRIOR ART)

System and process for enhancing method calls of special purpose object-oriented programming languages to have security attributes for access control

2/10

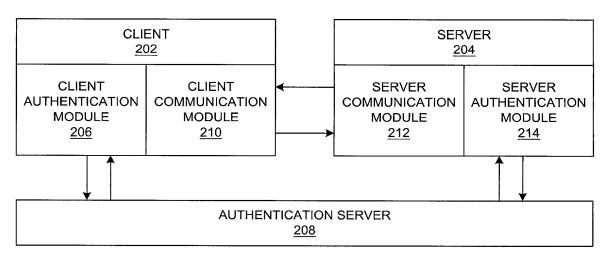
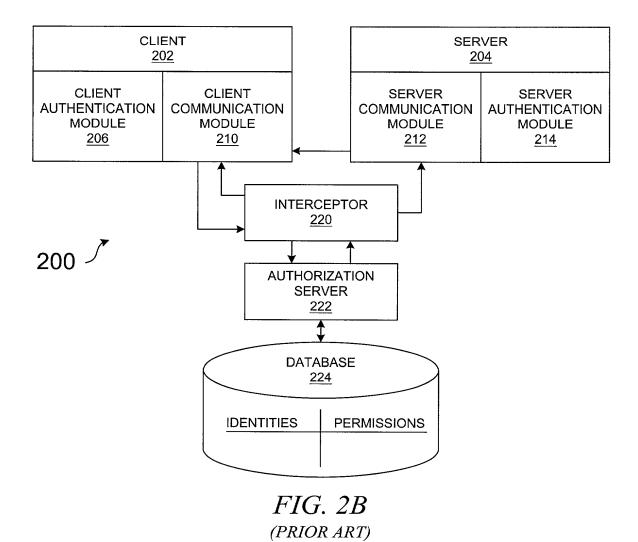


FIG. 2A(PRIOR ART)



System and process for enhancing method calls of special purpose object-oriented programming languages to have security attributes for access control

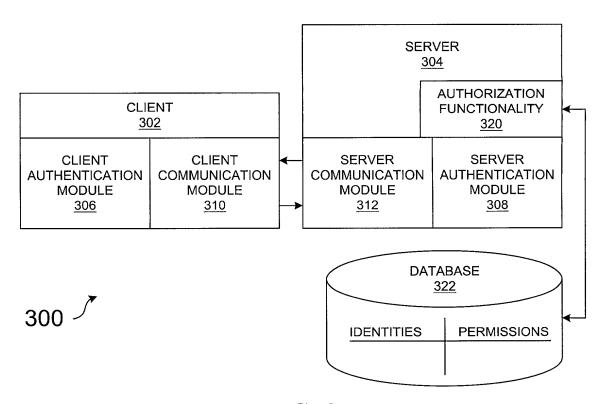


FIG. 3

System and process for enhancing method calls of special purpose object-oriented programming languages to have security attributes for access control

```
Class public ClientClassA {
               Method A {
                 try {
                       ServerClassM.M(); // Might throw an authorization exception.
                                            // Try block successfully executes when
                                            // invoked by entity with proper authority,
                                            // such as BankManager
402
                 catch (AuthException e)
                    // Codepath not reached for BankManager role
            Class public ClientClassB {
               Method B {
                 try {
                       ServerClassM.M(); // Might throw an authorization exception.
                                            // Try block fails to execute when invoked
                                            // by entity with improper authorization,
                                            // such as BankTeller
404
                  catch (AuthException e)
                    // Codepath reached for BankTeller role
            Class public ServerClassM {

    Method M ( ) enforces ManagerAuthorization.AuthMethod

            Class public ManagerAuthorization {
406
               Method AuthMethod ( ) throws AuthException {
                  // Body of AuthMethod containing authorization process
            }
```

FIG. 4A

System and process for enhancing method calls of special purpose object-oriented programming languages to have security attributes for access control

```
Class public ClientClassA {
             Method A {
                try {
                     ServerClassM.M(); // Might throw an authorization exception.
                                          // Try block successfully executes when
                                          // invoked by entity with proper authority,
                                          // such as BankManager role
402
                catch (AuthException e)
                  // Codepath not reached for BankManager role
             }
           Class public ClientClassB {
             Method B {
                try {
                     ServerClassM.M(); // Might throw an authorization exception.
                                           // Try block fails to execute when invoked
                                           // by entity with improper authorization,
                                           // such as BankTeller role
404
                catch (AuthException e)
                   // Codepath reached for BankTeller role
             }
           Class public ServerClassM enforces ManagerAuthorization.AuthMethod {
              Method M () {
           Class public Manager Authorization {
              Method AuthMethod ( ) throws AuthException {
             }
```

FIG. 4B

System and process for enhancing method calls of special purpose object-oriented programming languages to have security attributes for access control

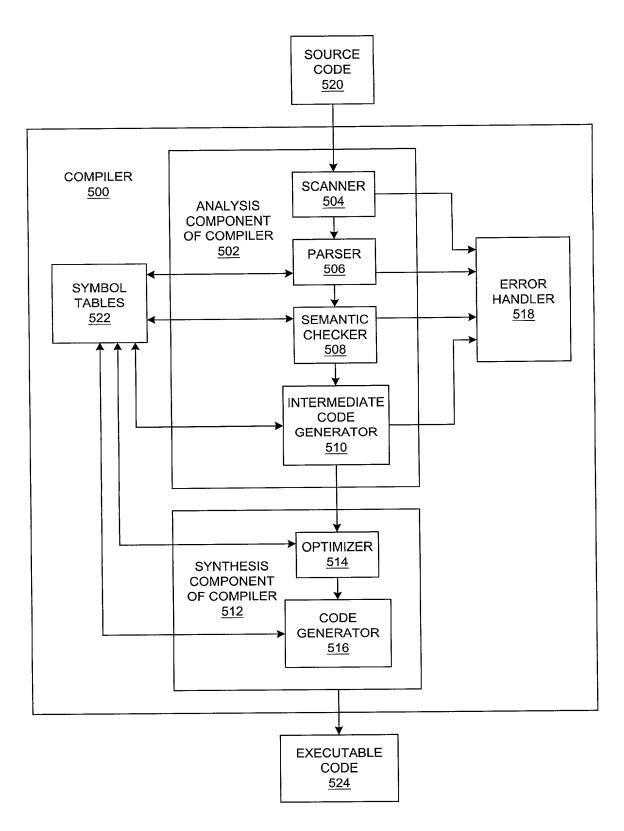


FIG. 5 (PRIOR ART)

System and process for enhancing method calls of special purpose object-oriented programming languages to have security attributes for access control

```
dec → classdec
classdec → class class_id extends class_id { { class_field } }
{\sf classdec} \rightarrow {\sf class\_id} \ {\sf extends} \ {\sf class\_id} \ {\sf enforces} \ {\sf authMethod} \ \{ \ {\sf class\_field} \ \} \} \blacktriangleleft 
                                                                                            602
class_field → vardec
class_field → method
method → method id(tyfields) enforces authMethod — 608
method → method id(tyfields) = exp
authMethod → method id(tyfields) = exp —— 604
method → method id(tyfields):type_id = exp
authMethod → method id(tyfields):type_id = exp — 606
exp → new class_id
\exp \rightarrow \text{lvalue id()}
exp \rightarrow Ivalue id(exp {, exp})
```

FIG. 6

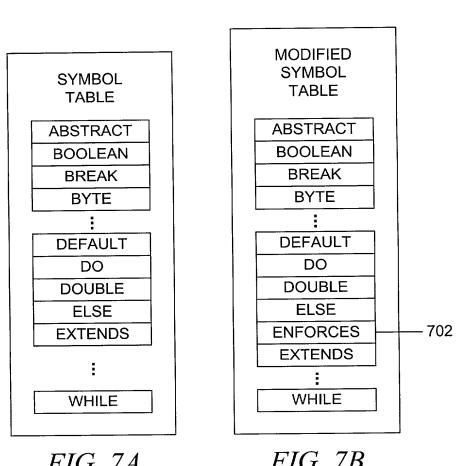


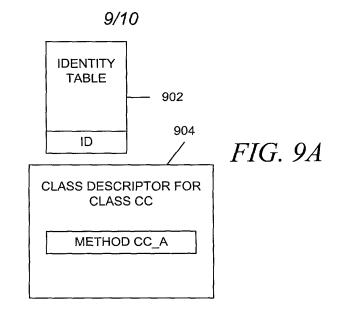
FIG. 7A

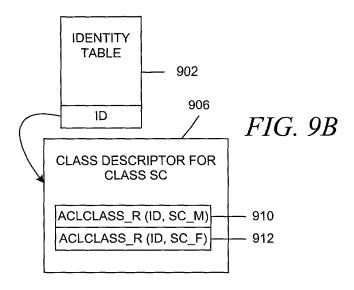
FIG. 7B

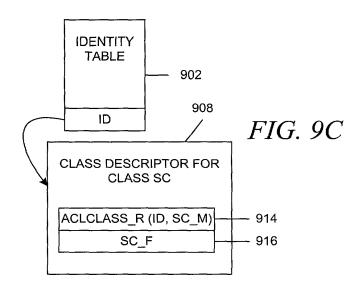
System and process for enhancing method calls of special purpose object-oriented programming languages to have security attributes for access control

```
8/10
802 — class CC extends Object {
              method A ( ) = { SC:M( ) }
        }
804 —— class SC extends Object enforces ACLClass:R {
              method M() = \{ body of M \}
              method F() = \{ body of F \}
        }
806 — class SC extends Object {
              method M ( ) enforces ACLClass:R = { body of M }
              method F() = \{ body of F \}
        }
808 —— class SCSC extends SC {
              method M() = \{ body of M \}
        }
                            FIG. 8
                               BEGIN
     SOFTWARE DEVELOPER SELECTS AN AUTHORIZATION METHOD FOR
             CONTROLLING ACCESS TO PROTECTED METHOD
                                1002
   SOFTWARE DEVELOPER USES ENFORCEMENT CONSTRUCT COMPRISING
  AUTHORIZATION METHOD AGAINST CLASS OR METHOD TO BE PROTECTED
                                1004
                      SOURCE CODE IS COMPILED
                                1006
                   EXECUTABLE CODE IS DEPLOYED
                                1008
                                END
                           FIG. 10A
```

System and process for enhancing method calls of special purpose object-oriented programming languages to have security attributes for access control







System and process for enhancing method calls of special purpose object-oriented programming languages to have security attributes for access control

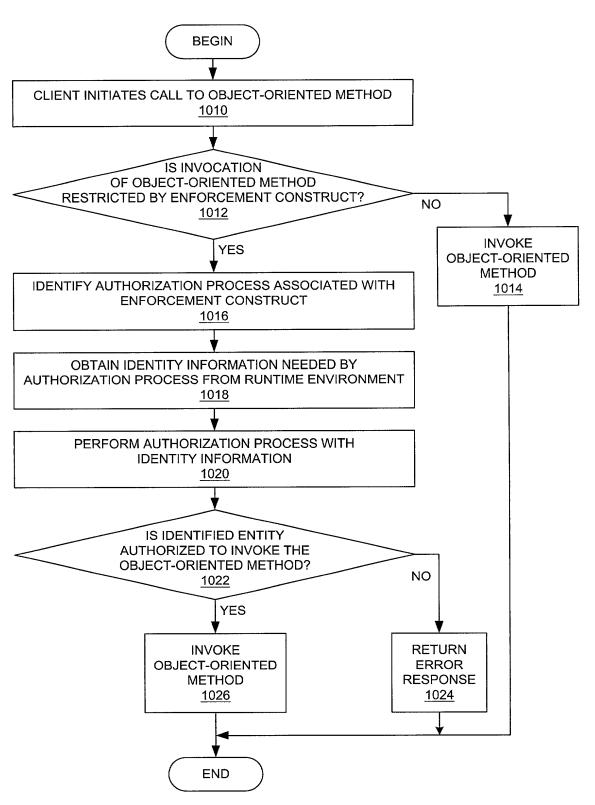


FIG. 10B